

# 5. Synthèse complète du projet Esportify

## Présentation générale

Le projet **Esportify** est une plateforme web de gestion d'événements e-sport, réalisée dans le cadre de l'ECF Développeur Web et Web Mobile.

Elle vise à couvrir, de manière concrète et modulaire, toutes les fonctionnalités attendues par le sujet d'examen :

- Inscription/gestion multi-rôles (visiteur, joueur, organisateur, administrateur)
- Création, validation et suivi d'événements en ligne (back-office)
- Gestion des jauges et participation en temps réel
- Modération, workflow d'approbation
- Classement et scoring des joueurs
- Espace communautaire de discussion (chat)
- Expérience utilisateur responsive, interface claire et accessible
- Documentation professionnelle et complète

Ce projet, mené **en autonomie**, a été découpé en modules pour permettre une intégration et un test progressif des fonctionnalités, avec une attention particulière portée à l'architecture générale (arborescence des dossiers, séparation front/back, modularité des scripts).

## Objectifs pédagogiques et techniques

L'objectif principal était de :

- **Maîtriser toutes les étapes du développement web** : analyse du besoin, modélisation des données, développement back-end, intégration front, sécurisation, tests et déploiement.
- **Mettre en œuvre une base de données relationnelle robuste**, en exploitant les jointures et les requêtes préparées.
- **Assurer la sécurité** (authentification, gestion de session, séparation des rôles, validation des entrées...).

- **Structurer le code** : chaque fonctionnalité dispose de son propre module JS et PHP pour une meilleure lisibilité, maintenance et évolutivité.
- **Développer une interface claire et accessible**, même si l'aspect esthétique a été relégué au second plan pour privilégier la logique applicative et la stabilité du back-end.
- **Produire une documentation exhaustive et professionnelle** (charte graphique, technique, gestion de projet, manuel utilisateur).

## Réponses détaillées au cahier des charges

### Gestion des rôles et des parcours

- **Visiteur** : accès à la vue globale des événements, filtrage dynamique, découverte des fonctionnalités, accès à la discussion publique.
- **Joueur** : inscription à un événement, historique des participations, visualisation du classement, participation à la discussion.
- **Organisateur** : création d'événements, gestion des inscriptions, attribution des points aux participants, workflow de validation.
- **Administrateur** : gestion complète des utilisateurs, modération du chat, validation/refus des événements, accès à toutes les données.

### Fonctionnalités principales

- **Tableau de bord dynamique** : affichage conditionnel selon le rôle, accès aux outils d'administration, visualisation des scores, participations, événements créés.
- **Filtrage avancé** : recherche par titre, jeu, statut, date, nombre de joueurs.
- **Workflow de validation** : événements "en attente" jusqu'à validation par l'admin, notifications intégrées.
- **Classement général et historique individuel** : calculs automatiques des scores, attribution selon le classement lors des événements.
- **Discussion communautaire (chat)** : zone d'échange entre utilisateurs, possibilité pour l'admin de modérer ou supprimer les messages.
- **Galerie dynamique** : carrousel d'images, mode admin pour l'ajout/suppression.
- **Notifications d'état** : jauges visuelles, bannières de statut, feedback utilisateur sur toutes les actions.

## Sécurité et fiabilité

- **Sessions PHP sécurisées** pour la gestion des droits.
- **Mot de passe hashé** systématiquement (password\_hash).
- **Requêtes SQL préparées (PDO)** sur tous les points sensibles.
- **Vérification systématique des rôles et des entrées**, côté serveur comme côté client.
- **Respect du RGPD** et des bonnes pratiques de gestion de la donnée.

## Base de données

- **Modèle relationnel** complet :
  - Table users (rôles, authentification, identité)
  - Table events (description, statut, auteur, jauge, image)
  - Table inscriptions (gestion participations)
  - Table scores (historique, classement)
  - Table messages (discussion)

## Démarche d'architecture, choix de développement et gestion du projet

- **Choix méthodologique** : développement incrémental, validation de chaque module avant passage au suivant (approche type agile).
- **Travail en autonomie**, appuyé sur une documentation continue, versionnage (Git), sauvegardes régulières, test manuel à chaque étape.
- **Organisation du code** : séparation stricte du front et du back, scripts nommés explicitement, modularité des controllers PHP, architecture "MVC simplifié".
- **Tests** : scénarios utilisateurs pour chaque profil, validation manuelle sur l'environnement local et AlwaysData en pré-déploiement.

## Limites, arbitrages et points d'amélioration

### Partie messagerie : choix du SQL

Le choix pour la **messagerie (chat)** s'est porté sur une solution basée sur la base relationnelle SQL (MariaDB) déjà en place, et non sur une technologie NoSQL (comme

MongoDB) qui aurait pu offrir plus de flexibilité et de performances en temps réel.

Ce choix est principalement lié au **manque de temps pour approfondir le sujet NoSQL** et à la volonté de garantir la cohérence de l'architecture globale.

La mise en œuvre d'un système NoSQL (MongoDB, Firebase, etc.) nécessitant une étude théorique et une intégration plus poussée n'a pas pu être réalisée dans les délais impartis.

## Sécurisation et déploiement

L'approche sécurité et déploiement a été limitée par l'accès aux modules de formation :

- **La partie théorique sur la sécurisation de l'infrastructure et des applications web n'a pas encore été abordée** à la date de rédaction de ce document (13/07 : début du module sécurité ; 18/07 : déblocage du module déploiement ; modules sécurité avancée prévus pour le 20 et 23/07).  
Les choix techniques réalisés le sont donc avec les connaissances acquises à date, et seront renforcés lors de la suite du parcours de formation.

## Architecture et support IA

Pour ce projet, j'ai fait le choix d'**exploiter pleinement les outils d'aide disponibles** (notamment ChatGPT), dans une démarche de **complément à ma propre réflexion et structuration du projet**.

L'IA a joué un rôle d'assistant, de vérificateur, d'explicateur et de support pour compenser mes lacunes, **mais j'ai assuré l'architecture, le pilotage, l'intégration et la validation de tous les choix**.

Ce choix, assumé et transparent, s'inscrit dans une démarche d'auto-formation responsable, où l'IA m'a permis de gagner du temps sur certains aspects opérationnels tout en m'aidant à comprendre l'ensemble des rouages d'un projet complet.

## Front-end, design et responsive

Dans le cadre d'un premier projet complet, **j'ai volontairement concentré mes efforts sur la partie back-end, la structure, la logique métier et la sécurité**, au détriment du front-end.

Le design est resté volontairement simpliste, le CSS basique et le responsive incomplet, afin de garantir avant tout la solidité, la sécurité et l'évolutivité du back-office.

Le projet est donc conseillé en usage desktop : une refonte UX/UI (et l'ajout d'un framework front moderne) pourrait être envisagée par la suite.

## Limites de compétences (JS, PHP, etc.)

Je ne maîtrise pas encore pleinement JavaScript ni PHP, l'essentiel de l'apprentissage a été fait "sur le tas" lors de ce projet, grâce à l'expérimentation et à l'accompagnement de l'IA.

C'est l'expérience, et la réalisation concrète de projets comme celui-ci, qui me permettra d'acquérir de réelles compétences et d'approfondir la maîtrise technique de ces langages.

## Conclusion et perspectives

Ce projet **constitue pour moi le premier véritable projet complet de développement web**, de l'analyse du besoin jusqu'à la documentation et au déploiement.

Il m'a permis :

- de comprendre et de mettre en pratique **l'ensemble du cycle de vie d'un projet web** (analyse, conception, modélisation, développement, tests, déploiement, documentation),
- d'expérimenter **la gestion multi-rôles, la sécurisation, la séparation front/back, le versionning et le travail en autonomie**,
- de produire une base solide sur laquelle je pourrai construire des projets plus ambitieux, tout en reconnaissant les limites actuelles et les axes d'amélioration prioritaires.

J'assume pleinement le recours à l'IA comme outil d'appui, dans un souci d'efficacité et de montée en compétence : **c'est en m'appuyant sur ces outils que j'ai pu acquérir une vision globale de l'architecture d'un projet, et que j'ai pu avancer rapidement sur les aspects techniques.**

Ce projet n'est qu'un **premier jet**, une fondation : il me reste encore du chemin à parcourir pour maîtriser toutes les technologies utilisées, mais cette expérience concrète m'a apporté une compréhension approfondie des enjeux et des bonnes pratiques du développement web moderne.

### Axes d'amélioration identifiés :

- Approfondissement de la sécurisation de l'infrastructure (modules à venir, intégration d'un reverse proxy, HTTPS, politique CORS...)

- Finalisation du responsive design (cards, dashboard, modales...)
- Ajout de tests automatisés (PHPUnit, JS)
- Exploration de solutions temps réel pour la messagerie (WebSocket, NoSQL, etc.)
- Amélioration de la documentation API et du process DevOps
- Mise en place d'un process d'intégration continue et de tests automatisés

Ce projet m'a permis de poser les bases solides de ma future progression : chaque étape, chaque choix, chaque difficulté rencontrée a enrichi ma compréhension du métier de développeur web et pose le socle de mon évolution professionnelle.